PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-093655
(43)Date of publication of application : 10.04.1998

(51)Int.Cl. H04L 29/04
G06F 13/00

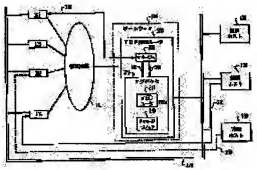
(21)Application number : 09-223516
(22)Date of filing : 20.08.1997

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>
(72)Inventor : CLEMENT R ATANAASHIO
GERMAN SERGIO GOLDSCHMIDT
GURNEY DOUGLAS HOROUEIIHANTO
STEPHEN EDWIN SMITH

(30)Priority
Priority number : 96 701939 Priority date : 23.08.1996 Priority country : US

(54) METHOD FOR DESIGNATING WITH ROUTE INCOMING MESSAGE AND SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To improve entire throughput by a gateway dividing EC into virtual ECs, dynamically distributing incoming connection within a virtual EC based on load of a current server according to a policy that can be structured and by supporting the dynamic structure of a cluster.
SOLUTION: A TCP connection router 300 is connected to a communication link of an interconnection 110 and connects with a computer (CP) nodes 105 to 109 which forms a cluster. One of CPs in a cluster which works as a gateway (GW) is connected to an external host (CP) through a network (NW) 120. The GW is connected to plural NW, also, plural nodes inside a cluster become GW, and the connection of each GW to a boundary can have plural addresses on NW. A manager 320 sends a command request 344 to an executor 340, evaluates response 346 and controls a route. Thereby, entire throughput can be improved.



LEGAL STATUS

[Date of request for examination]	30.07.1998
[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3452466
[Date of registration]	18.07.2003
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision	

of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号

特開平10－93655

(43)公開日　平成10年(1998)４月10日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｈ０４Ｌ　29/04		Ｈ０４Ｌ　13/00	３０３Ｂ
Ｇ０６Ｆ　13/00	３５５	Ｇ０６Ｆ　13/00	３５５

審査請求　未請求　請求項の数13　ＯＬ　（全 22 頁）

(21)出願番号　特願平9－223516

(22)出願日　平成９年(1997)８月20日

(31)優先権主張番号　０８／７０１９３９

(32)優先日　1996年８月23日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　390009531
インターナショナル・ビジネス・マシーンズ・コーポレイション
INTERNATIONAL BUSINESS MASCHINES CORPORATION
アメリカ合衆国10504、ニューヨーク州　アーモンク　（番地なし）

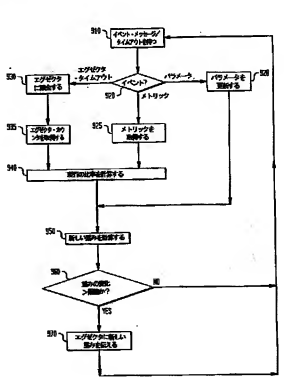
(72)発明者　クレメント・アール・アタナーシオ
アメリカ合衆国10566　ニューヨーク州ピークスキル　ディルッポ・ドライブ　５

(74)代理人　弁理士　坂口　博　（外１名）

最終頁に続く

(54)【発明の名称】　着信メッセージを経路指定する方法及びシステム

(57)【要約】
【課題】　カプセル化クラスタの全体的なスループットを改善すること及び遠隔サービス要求の総合的な遅れを少なくすること。
【解決手段】　ＴＣＰ接続ルータは各カプセル化クラスタをいくつかの仮想ＥＣ（ＶＥＣ）に分割し、構成可能なポリシーにしたがって現行のサーバ負荷メトリックに基づいてＶＥＣ内に着信接続を分配することによってカプセル化クラスタリングを行う。実施の形態の１つにおいて、接続ルータはクラスタの動的な構成をサポートし、ＶＥＣクライアントに中断のないサービスを与える透過的回復を可能とする。



1

【特許請求の範囲】

【請求項１】コンピュータ・ノードのクラスタの境界を越えて着信メッセージを経路指定する方法であって、前記クラスタが１つまたは複数のネットワークに結合されており、前記方法がポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号及び目標アドレスを見つけだし、読み取るステップと、

前記目標アドレスに基づいて、前記コンピュータ・ノードのサブセットを選択するステップと、

前記ポート番号に基づいて、前記サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定する機能を選択するステップと、

前記メッセージを前記経路の宛先に送るステップと、

前記メッセージが境界を越えて送られている間に、前記サブセット内のメンバシップの少なくとも１つと前記サブセットの数を動的に変更するステップとを備えている方法。

【請求項２】選択が前記ポート番号と前記メッセージ・ヘッダ内のプロトコル識別子に基づいていることを備えている、請求項１に記載の方法。

【請求項３】前記サブセットの数が監視機能によって動的に変更される、請求項１に記載の方法。

【請求項４】前記サブセットのメンバシップが監視機能によって動的に変更される、請求項１に記載の方法。

【請求項５】変更が前記サブセットのメンバの置換、及び前記サブセットへのメンバの追加の少なくとも１つを含む、請求項３に記載の方法。

【請求項６】コンピュータ・ノードのクラスタを越えて着信メッセージを透過的に経路指定するシステムであって、前記クラスタが１つまたは複数のネットワークに結合されており、前記システムがポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号及び目標アドレスを見つけだし、読み取って、前記コンピュータ・ノードのサブセットを選択する手段と、

前記ポート番号に基づいて、機能を選択し、該機能が前記サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定し、該経路の宛先が前記サブセット内の前記コンピュータ・ノードである手段と、

前記サブセット内のメンバシップの少なくとも１つと前記サブセットの数を動的に変更する手段とを備えているシステム。

【請求項７】前記サブセットの数が監視機能によって動的に変更される、請求項６に記載のシステム。

【請求項８】前記サブセットのメンバシップが監視機能によって動的に変更される。請求項７に記載のシステム。

【請求項９】変更が前記サブセットのメンバの置換、及び前記サブセットへのメンバの追加の少なくとも１つを含む、請求項７に記載のシステム。

【請求項１０】コンピュータ・ノードのクラスタの境界

2

を越えて着信メッセージを経路指定する方法であって、前記クラスタが１つまたは複数のネットワークに結合されており、前記方法が境界ノードにおいて、

ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号を見つけだし、読み取るステップと、

前記ポート番号に基づいて、サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定し、該経路の宛先が前記クラスタ内のコンピュータ・ノードであるステップとを備えており、

前記境界ノードの障害を検出するステップと、

障害の検出に応答して、状態情報のサブセットを前記クラスタ内の各ノードから代替境界ノードへ転送するステップと、

前記代替境界ノードにおいて、

前記クラスタ内のノードから状態情報のサブセットを収集するステップと、

前記状態情報を使用して、障害前の前記境界ノードの作動状態を再構成し、メッセージが障害前に前記境界ノードによって達成されていたのと同じ態様で、前記代替境界ノードによって分配されるようにするステップとを備えている方法。

【請求項１１】ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号及び目標アドレスを見つけだし、読み取るステップと、

前記目標アドレスに基づいて、前記コンピュータ・ノードのサブセットを選択するステップとをさらに備えており、

経路の宛先が前記サブセットから選択される、請求項１０に記載の方法。

【請求項１２】コンピュータ・ノードのクラスタの境界ノードの障害から回復するシステムであって、

前記境界ノードにおいて、ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号を見つけだし、読取り、前記ポート番号に基づいて、サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定する機能を選択し、前記経路の宛先が前記クラスタ内の前記コンピュータ・ノードである手段と、

前記境界ノードの障害を検出する手段と、

障害の検出に応答して、前記クラスタ内の各ノードから状態情報のサブセットを収集する手段と、前記サブセットから、障害前の前記境界ノードの作動状態を再構成し、メッセージが障害前に前記境界ノードによって達成されていたのと同じ態様で、代替ノードによって分配されるようにする手段とを備えている代替境界ノードとを備えているシステム。

【請求項１３】ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号を見つけだし、読取り、前記ポート番号に基づいて、サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定する機能を選択し、前記経路の宛先がクラスタ内のコンピュー

タ・ノードである手段と、
障害の検出に応答して、前記クラスタ内の各ノードから状態情報のサブセットを収集する手段と、サブセットから、障害前の境界ノードの作動状態を再構成し、メッセージが障害前に前記境界ノードによって達成されていたのと同じ態様で、代替ノードによって分配されるようにする手段とを備えているコンピュータ・ノードのクラスタで使用される境界ノード。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明はネットワーク・コンピューティングに関する。詳細にいえば、本発明はコンピュータをクラスタ化して一連の遠隔サービスをサポートすることに関する。

【０００２】

【従来の技術】カプセル化クラスタ（ＥＣ）は一連のサービス（たとえば、ウェブ・サービス、ＮＦＳなど）をもたらす接続ルータ（ＣＲ）と複数のサーバ・ホストを特徴としている。カプセル化クラスタリングを提供するシステムの例には、米国特許第５３７１８５２号がある。

【０００３】遠隔クライアントは、たとえばＴＣＰ／ＩＰ（たとえば、ＨＴＴＰ）に基づくプロトコルを使用してＥＣにサービスを要求する。各要求に対するサービス時間はサービスのタイプ、及び対応するサーバ・アプリケーションの可用性に応じて変動する。それ故、接続の固有割振りは、利用可能なＥＣリソースの利用度が低い偏った割振りを急速に作り出し、要求に対して不必要な遅れを導入する。

【０００４】従来技術はサーバのスケーリングに関連する多くの性能上の問題が存在していることを示している。たとえば、ＮＣＳＡの「World Wide Web Server: Designand Performance」、IEEE Computer、第２８巻第１１号、１９５１１月、６８－７４ページ参照。ウェブ・サーバ（すなわち、ｈｔｔｐｄデーモン）をサポートするのにラウンドロビンＤＮＳを使用しているＥＣを考えてみる。サーバはｈｔｔｐを介して、ビデオ・ストリーム、データベース照会、及び静止ウェブ・ページへのアクセス・サービスを与える。各タイプの要求に対するサービス時間は、与えられるサービスのタイプ、及び関与する実際のコンテンツに応じて変動する。たとえば、複雑なデータベース照会には静止したプリロードＨＴＭＬページの何倍もの時間がかかる。要求処理時間におけるこの不釣り合いはサーバ・クラスタの偏った利用を引き起こすことがしばしばある。ラウンドロビンＤＮＳに関連した問題は、「User Access Patterns to NCSA's Worldwide Web Server」、Kwan他、Technical Report UIUCDSD-R-95-1394, Department of Computer Science, University of Illinois Urbana-Champaign, １９９５年２月に記載されている。

【０００５】従来技術はリソースの動的割振りの必要があることを示している。たとえば、「Evaluating Management Decisions via Delegation」、German Goldszmidt and Yechiam Yemeni, The Third International Symposium on Integrated Network Management, San Francisco, CA, １９９３年４月を参照されたい。ＥＣは通常はホストの集合体に対して、サーバの集合体の単一のシステム・イメージを与える。しかしながら、実際のインストレーションはサービスを特定のユーザのポリシーにしたがって割り振ることを必要とすることがある。たとえば、ホストの特定のサブセットをマーチャント・ウェブ・サーバの機密トランザクションに割り振るとともに、ビデオ・オン・デマンド・サービスを、専用のハードウェアを含んでいるサーバの他のサブセットによってサポートする。

【０００６】

【発明が解決しようとする課題】本発明の目的は、カプセル化クラスタの全体的なスループットを改善することである。

【０００７】本発明の他の目的は、遠隔サービス要求の総合的な遅れを少なくすることにある。

【０００８】本発明のさらに他の目的は、指定されたノードが障害を起こした接続ルータの動作を引き継ぎ、ネットワーク・クライアントがサービスの中断を受けないようにする手段を提供することである。

【０００９】

【課題を解決するための手段】本発明の第１の態様によれば、カプセル化クラスタ（ＥＣ）はゲートウェイ・ノードとサーバ・ホストとを特徴としている。ゲートウェイ・ノードは（１）ＥＣをいくつかの仮想ＥＣ（ＶＥＣ）に分割し、（２）構成可能なポリシーにしたがって現行のサーバの負荷に基づいて着信接続をＶＥＣ内に動的に分配し、（３）クラスタの動的な構成をサポートする。

【００１０】本発明の第２の態様によれば、ゲートウェイ・ノードの障害からの透過的な回復を可能として、クライアントへの中断のないサービスを提供するシステム及び方法が提供される。この方法によれば、クラスタまたはＶＥＣ内の各ノードはゲートウェイで保持されている状態情報のサブセットのコピーを維持する。ゲートウェイが障害を起こすと、状態情報がバックアップ・ゲートウェイへ転送される。

【００１１】好ましい実施の形態において、ＥＣは（１）ＶＥＣ（すべての遠隔クライアントに対して単一のＩＰアドレス）、または（２）複数のＶＥＣ（いくつかのＩＰアドレスをエイリアス化する）として現れる。ＴＣＰ－ＣＲノードはＩＰアドレスを所有しており、すべてのこれらのＴＣＰ接続要求を受け取る。各ＩＰアドレスはＶＥＣと関連づけられる。ＴＣＰ－ＣＲはＶＥＣに関連づけられた重みにしたがって、新しいＴＣＰ接続

5　　6

をホストへ分配する。TCP－CRはVECの動的定義、VECに関連づけられた重みの動的構成、VECの自動または手動管理（ホスト、サービスなどの追加、除去）を可能とする。この解決策はサーバ・ホストの動的構成、追加または除去を可能とするとともに、ネットワーク内のキャッシュされたサーバ名の問題を回避する。

【0012】

【発明の実施の形態】

1．概要

本仮想カプセル化クラスタ・システムは米国特許第5371852号の改良形として実施できる。米国特許第5371852号は参照することにより、以下に全文が記載されているかのように、本明細書の一部となるものである。図1は米国特許第5371852号のカプセル化クラスタの発明の実施の形態を示している。米国特許第5371852号のシステムと同様に、本システムはTCP情報を送り、これはコンピュータ・クラスタの境界を越える。情報はポート・タイプ・メッセージの形態をしている。着信メッセージが送られ、サーバが応答するので、各クラスタは外部ホストに対して単一のコンピュータ・イメージとして現れる。本システムにおいて、クラスタはいくつかの仮想クラスタ（仮想カプセル化クラスタ）に分割されている。各仮想カプセル化クラスタは、クラスタ外にあるネットワーク上の他のホストに対して単一のホストとして現れる。メッセージは負荷のバランスがクラスタ・ノードのセットの間で維持されている態様で、各仮想カプセル化クラスタのメンバに送られる。

【0013】図3はTCPファミリのプロトコル用の接続ルータ、すなわちTCP接続ルータ（TCP－CR）300の実施の形態を示す。この装置は相互接続110と呼ばれる通信リンクによってまとめて接続され、クラスタを形成している2つ以上のコンピュータ・ノード（105－109）を備えている。（本発明の実施の形態の1つにおいて、相互接続はネットワークとなることに留意されたい。）ゲートウェイ109として働く、クラスタ内のコンピュータの1つはネットワーク120と呼ばれる他の通信リンクを介して、1つまたは複数の外部コンピュータもしくはクラスタ（ホスト）に接続されている。ゲートウェイを2つ以上のネットワークに接続することができ、またクラスタ内の2つ以上のノードがゲートウェイになることができる。ネットワーク、すなわち境界への各ゲートウェイの接続はネットワーク上で複数のアドレスを有することができる。各ゲートウェイはマネージャ320及びエグゼクタ340からなるTCP接続ルータ（TCP－CR）300と、図12に記載する光学式回復マネージャとを有している。マネージャはコマンド要求344をエグゼクタへ送り、応答346を評価することによって、経路を制御する。エグゼクタは米国特許第5371852号のものと同様なメッセージ・スイッチ140及びVECルータ310からなっている。

【0014】図4は本発明の他の実施の形態を示す。好ましい実施の形態におけるのと同様に、クラスタのノード107はその応答を直接クライアント130へ返送する。しかしながら、この実施の形態においては、専用の相互接続110（図3におけるような）はなく、すべてのクラスタ・ノードが外部のネットワーク120によって接続されている。TCP接続ルータは同じものである。サンプル要求348はゲートウェイ109を通ってクライアント130から、外部ネットワーク120を介してクラスタ・ノード107へ送られる。対応する応答350はノード107からクライアント130へ外部ネットワーク120を介して直接送られる。

【0015】マネージャ320の構成要素は接続割振りポリシーを実施し、仮想カプセル化クラスタの動的構成を可能とする。マネージャは動的フィードバック制御ループを介して各カプセル化クラスタのメンバにおける現行の負荷を監視し、評価する。マネージャは接続割振りポリシーを実施するものであり、このポリシーは仮想カプセル化クラスタ・サーバ全体にわたる着信接続のインテリジェントな拡散を行って、クライアント要求のサービスを迅速化する。新しい重みの割当てが、クラスタ・アドミニストレータによって構成できるマネージャ・アルゴリズムによって計算される。重み割当てに対するこの判断アルゴリズムの入力は、評価された負荷メトリック、及び時間閾値などのアドミニストレータが構成できるパラメータを含んでいる。着信接続は上記の入力に基づいて各VECに動的に割り振られ、クラスタ・リソースがクライアントに最高速のサービスを与えるように割り振られるようにする。マネージャはコマンド・インタフェースも含んでおり、このインタフェースは仮想カプセル化クラスタを動的に構成するために、アドミニストレータによって使用される。マネージャの詳細については、第3節で説明する。

【0016】TCP接続ルータ109が動作を停止した場合、クラスタのすべてのノードはそれぞれの遠隔クライアントにサービスを行うことができなくなる。この問題に対処するため、機能しているゲートウェイが障害を起こした場合に、指定のバックアップ・ゲートウェイ・ノード内で活動状態となり、かつサーバ・ノードを拡張して、回復データを維持するようにする回復マネージャを追加した。クライアントがゲートウェイの障害から回復するために何らかの処置を取る必要はなく、クラスタから中断なしにサービスを受け取る。回復マネージャの詳細については、第4節で説明する。

【0017】2．エグゼクタ

図5はエグゼクタ340の好ましい実施の形態を示す。エグゼクタはコマンド・プロセッサ540、メッセージ・スイッチ140、及びVECルータ310からなる。

コマンド・プロセッサ５４０はエグゼクタ３４０に対する要求を受け取って、応答３４６を返す。コマンド・プロセッサはメッセージ・スイッチ１４０及びＶＥＣルータ３１０と対話を行って、要求を行い、応答を構成する。コマンド・プロセッサは接続テーブル５１０、ＶＥＣテーブル５５０、ポート・テーブル５２０またはサーバ・テーブル５３０に影響を及ぼす。メッセージ・スイッチ１４０及び接続テーブル５１０は、米国特許第５３７１８５２号のメッセージ・スイッチ及び接続テーブルと同じものである。本発明の好ましい実施の形態において、ＶＥＣルータ３１０は着信パケットを修正しない。パケットはサーバへ送られるが、サーバは応答が内部ノードからクライアントへ直接送られるように構成されている。

【００１８】メッセージ・スイッチ１４０は本質的に、米国特許第５３７１８５２号のメッセージ・スイッチと同じものである。しかしながら、本発明のため、好ましい実施の形態のメッセージ・スイッチは最適化され、付加的なチェックがメッセージ・スイッチに追加されている。メッセージ・スイッチはメッセージがＶＥＣルータに既知のＶＥＣに対するものであるかどうかを調べるためにチェックを行わなければならない。

【００１９】ＶＥＣルータは外部ネットワーク上のクライアントに対する各ＶＥＣを表す一連のアドレスを維持している。ＶＥＣルータは要求をクラスタの内部ノードへ送るが、受け取った要求を修正しない。クラスタの各内部ノードは１つまたは複数のＶＥＣに関連づけられており、関連づけられているＶＥＣに対する要求だけを受け取る。当分野で周知の技法を使用して、本発明においては、ＶＥＣを表すアドレスに送られたパケットを受け入れ、クライアントに直接応答するように、内部ノードを構成する。従来技術においては、メッセージ・スイッチ１４０は着信要求（図１の１４０）に合わせてパケット・ヘッダを書き換え、要求に対する応答（図１の１２０）に合わせてパケット・ヘッダを書き換えなければならなかった。本発明においては、パケット・ヘッダの書き換えは必要ない。（従来技術を本発明とともに使用することはできる。）パケット・ヘッダが書き換えられず、応答パケットがゲートウェイ・ノード１０９を通って流れないため、本発明の性能は従来技術よりも良好なものである。応答パケットがＴＣＰ接続ルータを通って流れないため、メッセージ・スイッチはクラスタ内部のノードから応答パケットを取ることはない。その結果、好ましい実施の形態において、ヘッダの書き換えが排除され、内部ノードからの応答パケットに対するチェックが排除される。

【００２０】この改善の直接的な結果として、ＶＥＣが調べるのはクライアントと、サービスを与える内部ノードとの間の流れの半分だけとなる。これは正確な接続テーブルを維持することを困難とする。この問題を解決するために、本発明はその接続テーブルに固有な２つの新しいタイマ、状態タイムアウトとＦＩＮタイムアウトを使用する。これら２つのタイマと通信流、及び当分野で周知のタイマを使用すると、接続テーブルを正確に維持することができる。

【００２１】接続テーブル項目は２つの状態ＡＣＴＩＶＥ及びＦＩＮの１つであると考えられる。新しい接続が確立された場合には、接続テーブル項目が作成され、活動状態にされる。パケットが接続テーブル内に項目がある接続を流れている場合、接続項目にはタイム・スタンプがつけられる。ＶＥＣルータがクライアントからサービスを与えるノードへのＦＩＮ流を見つけた場合、関連する接続テーブル項目がＦＩＮ状態にされる。（パケットはＦＩＮ状態にされた接続上を流れることができる。）接続テーブル項目は閉じられており、最後のパケットがクライアントからその接続上のサーバに送られてから、ＦＩＮタイム・アウトによって特定されている時間数が経過した場合に、バージを行うのに利用できると考えられる。クライアントがＦＩＮを送出することなく障害を起こした場合、接続レコード項目が残る。失効したタイムアウトは最後のパケットが活動会話上を流れてからどれくらい待ってから、接続テーブル項目をバージするかを指定する。

【００２２】図７－図９はＶＥＣルータ３１０の流れ図を示す。図７において、ＶＥＣルータはパケット７０２を待っている。パケットを受け取ると、ＶＥＣルータはパケットが既存のＴＣＰ接続に対するものであるか、新しいＴＣＰ接続に対するものであるかをチェックする（７４０）。パケットが既存のＴＣＰ接続に対するものである場合、パケットがＦＩＮであるか、ＳＹＮであるか、ＲＳＴであるか（すべてのパケット・タイプは当分野で周知のものである）をチェックする（７０８）。パケットがこれらのもの１つでない場合には、パケットを接続に関連づけられた内部ノードに送る（７２２）。これ以外の場合には、パケットがＲＳＴであるかどうかをチェックする（７０１）。パケットがＲＳＴである場合には、会話が接続テーブルからバージされて、接続をリセットし（７１２）、パケットはその接続と関連づけられている内部ノードに送られる（７２２）。パケットがＲＳＴでない場合には、ＶＥＣルータ３１０はパケットがＳＹＮであるかどうかをチェックする（７１４）。パケットがＳＹＮである場合には、接続を確立し（７１６）、この接続が以前に存在していても、その接続を活動状態にする。ＶＥＣルータ３１０は次いで、パケットがＦＩＮであるかどうかをチェックする（７１８）。パケットがＦＩＮである場合には、接続はＦＩＮ状態となる（７２０）。ＦＩＮ処理後、あるいはパケットがＦＩＮでない場合、パケットは接続に関連づけられているサーバへ送られる（７２２）。

【００２３】図８は非存在接続の流れ図を示す。チェッ

ク７０４で非存在接続が見つかった場合、ＶＥＣルータはまずパケットがＳＹＮであるかどうかをチェックする（７２４）。パケットがＳＹＮでなければ、廃棄する（７２６）。パケットがＳＹＮである場合、接続が活動状態にセットアップされ（７２８）、サーバが選択され（７３０）、パケットが選択されたサーバへ送られる（７２２）。

【００２４】図９は新しい接続に対してサーバを選択する（７３０）処理の流れ図を示す。本発明において、この機能は加重経路指定を実現する。サーバの選択に関するこの検討のために、ＶＥＣの内部ノードには１からｎまでの番号がつけられているものと考える。たとえば、ＶＥＣに７つのノードがある場合、番号は１、２、３、４、５、６及び７となる。サーバの選択に関するこの検討のために、適格な重みを適正な最大値から１までの番号がつけられているものと考える。たとえば、最大適正値が５である場合、適格な重みは５、４、３、２、及び１となる。０は特別な値である。重みは降順で選択することもできる。本発明は重みを個々ののサーバを与える各内部ノードと関連づける。これは各サービスに対して、少なくとも１つのノードが最大の非ゼロの重みを有しているか、あるいはすべてのノードがゼロの重みを有していることを保証する。

【００２５】サーバを選択する機能（７３０）はまず、次にもっとも高いサーバ（７３４）と現行の適格な重みを拾い出す。次いで、この値が大きすぎるものかどうかをチェックする（７３５）。この値が大きすぎない場合には、この値に対応するサーバが良好な選択であるかどうかをチェックする（７４６）。（このチェックについては、後述する。）この値が大きすぎる場合には、最初のサーバ（７３６）と次に低い重みを拾い出す。次いで、次に低い重みがゼロであるかどうかをチェックする（７３８）。次に低い重みがゼロでない場合には、現行の適格な重みの代わりに、これを使用する。この機能は現行のサーバが良好な選択であるかどうかをチェックする（７４６）。最初のサーバと最大の重みを選択した後、この機能はパケットを送るのに利用できるサーバがあるかどうかをチェックする（７４２）。利用可能なすべてのノードの重みがゼロである場合、サーバはまったく利用できない。利用できるサーバがない場合には、サーバを選択することなく、パケットは戻される（７４４）。利用可能なサーバがある場合には、この機能は良好な選択があるかどうかをチェックする（７４６）。良好な選択は、重みが現行の適格な重みよりと等しいか、それよりも大きいサーバとして定義されている。これが良好な選択である場合には、サーバが選択され（７４８）、ＶＥＣルータへ戻される（７５０）。これが良好な選択でない場合には、アルゴリズムは次のサーバを拾い出す（７３４）。

【００２６】最大の重みが非ゼロであり、少なくとも１つのノードが最大の重みを有しているか、あるいはすべてのノードの重みがゼロであるため、サーバ選択機能は常に終了する。正の重みのあるノードがある場合、サーバを選択する機能は重みの比に基づいてパケットを分配する。たとえば、任意の２つのノードの間で、一方の重みが３であり、他方の重みが２である場合、重みが３のノードは、重みが２のノードが受け取る２つのパケットごとに３つのパケットを取得する。

【００２７】図１０はＶＥＣルータが使用するデータ構造の実施の形態を示す。ＶＥＣテーブル５５０は外部ネットワーク上のＶＥＣアドレスである一連のアドレスを含んでいる。ＶＥＣに個々に関連づけられるすべてのパラメータも、このテーブルに含まれている。各ＶＥＣテーブルはポート・テーブル５２０に関連づけられており、このポート・テーブルはＶＥＣがサービスを与えている一連のポート８０２を含んでいる。各ポート項目８０２には失効タイムアウト８０４、ＦＩＮタイムアウト８０６及びその他のポート固有の属性８０８が関連づけられている。各ポートには、このポートに関連づけられているサービスを与えるために使用されるＶＥＣの内部ノードのサブセットが関連づけられている。ノード・テーブル５３０はポートに関連づけられているノード８２０のアドレス、このノードに関連づけられている現行の重み８２２、及びその他のノード固有の情報８３０を含んでいる。（ノード固有の情報の例は、活動状態の接続の数、ＦＩＮ状態の接続の数、及び完成した接続の総数を示すカウンタである。）ノード・テーブル５３０はこのテーブル内のノードのセットに対して加重経路指定を実施するためにサーバを選択する機能に必要な状態も含んでいる。ノード・テーブルはノード８１０の総数、最後に選択されたノード８１２、現行の適格な重み８１４、最大重み８１６、及び重みのバウンド８１８を含んでいる。重みのバウンドは最大重みの変動を制限するために使用される。いずれのノードも、重みのバウンドよりも大きい重みを持つことはできない。

【００２８】３．マネージャ

接続ルータマネージャ（マネージャ３２０）の発明は構成可能なポリシーにしたがって、いくつかの負荷メトリックを使用して着信接続を動的に分配する方法及び装置である。マネージャはエグゼクタ３４０の経路指定アルゴリズムの重みを動的に修正して、クラスタ・リソースの割振りを最適化する制御ループを備えている。本発明の目的は、クラスタの現行状態にしたがって着信ＴＣＰを分配することによって、クラスタの全体的なスループットを改善し、サービス要求の総合的な遅れを少なくすることである。それ故、本発明は接続をサーバ・ホストへ分配する、サーバの利用度を改善し、要求のサービスを行うことの遅れを少なくする方法を記載する。

【００２９】図６は５つのノード（１０５、１０６、１０７、１０８、及び１０９）のクラスタ６００内の、本

発明のマネージャ３２０の実施の形態のサンプルを示す。図６は図４の代替ネットワーク構成を使用しているが、図３の構成も可能である。ノードの１つはゲートウェイ１０９であり、これは外部ネットワーク１２０に接続しており、ＴＣＰ接続ルータ３００（エグゼクタ３４０及びマネージャ３２０）を実行する。マネージャ３２０は５つの一般的な構成要素、負荷マネージャ（Ｍｂｕｄｄｙ）６１０、外部制御インタフェース（Ｃａｌｌｂｕｄｄｙ）６２０、クラスタ・ホスト・メトリック・マネージャ（Ｈｏｓｔｍｏｎｉｔｏｒ）６３０、順方向メトリック・ジェネレータ（ＦＭＧ）６４０、及びユーザ・プログラマブル・メトリック・マネージャ（ＵＰＭＭ）６５０からなっている。

【００３０】Ｍｂｕｄｄｙ６１０は入力メトリック、ホスト・メトリック、サービス・メトリック、及びユーザ・メトリックという４つの異なるクラスのメトリックを使用して、エグゼクタ用の重み機能を実行することができる。Ｍｂｕｄｄｙ６１０はこれらのメトリックとその他の関連する情報を、エグゼクタ・インタフェース３４６、Ｃａｌｌｂｕｄｄｙインタフェース６２４、Ｈｏｓｔｍｏｎｉｔｏｒインタフェース６３４、ＦＭＧインタフェース６４４、及びＵＰＭＭインタフェース６５４から受け取る。Ｍｂｕｄｄｙはインタフェース３４４を介して、各ＶＥＣポート・サーバに対するエグゼクタ経路指定アルゴリズムに関連づけられた重みを制御する。

【００３１】Ｍｂｕｄｄｙ６１０は定期的にエグゼクタ３４０に、インタフェース３４６を介して、各サーバに関連づけられた内部カウンタの値を要求する。たとえば、各サーバに対して確立された接続の総数に関するカウンタの値を定期的に要求する。時間Ｔ１及びＴ２でポーリングされた２つのカウンタを減算することにより、Ｍｂｕｄｄｙ６１０は時間間隔Ｔ１－Ｔ２中に受け取った接続の数を表すメトリック変数を計算する。このような入力メトリックの集合は各ＶＥＣ及び各ポート・サービスに対する接続要求の特性速度の概数を与える。

【００３２】Ｈｏｓｔｍｏｎｉｔｏｒ６３０は定期的にＭｂｕｄｄｙ６１０に、メッセージ・インタフェース６３４を介して、クラスタ内の各ホストの状態に関する情報を送る。この状態情報を得るのには、さまざまな周知の方法がある。たとえば、Ｈｏｓｔｍｏｎｉｔｏｒはプログラム・スクリプトを実行して、ホスト固有のメトリックを評価する監視エージェント６３５を使用することができる。たとえば、スクリプトはネットワーク接続に対するメモリ・バッファの現行レベルの利用度を評価することができる。メトリック・レポートをポリシー固有の閾時間内に受け取らなかった場合には、対応するホスト・メトリックに特別な値が与えられ、マネージャはホストに到達できないと判断することができ、それ故、それ以上の接続要求はこのホストに送られない。Ｈｏｓｔｍｏｎｉｔｏｒ６３０はすべての監視エージェントのレポートを調整し、これをＭｂｕｄｄｙに提示する。

【００３３】順方向メトリック・ジェネレータ（ＦＭＧ）６４０は順方向要求、すなわちゲートウェイ１０９のコンピュータから始まる要求を使用して、アプリケーション固有のメトリックまたはサービス固有のメトリックを作成し、評価する。評価はクラスタ・ホスト・サーバの各々に対して適切な要求を作成し、これらの応答の遅れを測定することからなっている。たとえば、ＨＴＴＰサーバにおける順方向遅延メトリックを取得するために、ＦＭＧは特定のポート（たとえば、ポート８０）にサービスを行うクラスタにおける各ＨＴＴＰサーバに対してＨＴＴＰ「ＧＥＴ　／」を生成する。次いで、ＦＭＧ６４０はＨＴＴＰ要求にサービスを行う際の対応する遅れを測定し、メトリックス・ベクトルをＭｂｕｄｄｙ６１０へ送る。ポリシー固有の閾時間までに要求に回答がない場合には、ＦＭＧは対応するサービス・ノードに、特定のサービス・タイプの新しい要求を一時的に受け取らないとのマークをつける。この情報をマネージャが使用して、特定のホストにおけるサービスに一時的に到達できないと判断し、それ故、このタイプのこれ以外の接続要求はこのサービスに送られなくなる。

【００３４】ユーザ・プログラマブル・メトリック・マネージャ（ＵＰＭＭ）６５０により、本発明のユーザが任意の新しいメトリックを接続の管理のために考えることを定義できるようになる。このようなメトリックは所与のクラスタ・インストレーションが実施を望むことのできる任意のポリシーを記述できる。たとえば、任意のポリシーは管理上の考慮事項のため、ある期間中にあるセットのクラスタ・ホストがＴＣＰ接続を受け取ってはならないことを必要とすることができる。ＵＰＭＭ６５０はインタフェース６５４を介して、メトリックとしてこれらのポリシーをＭｂｕｄｄｙに通信する。

【００３５】Ｃａｌｌｂｕｄｄｙ６２０の構成要素はアドミニストレータが、Ｍｂｕｄｄｙ６１０のパラメータの任意のものを動的に調節することを可能とする。Ｃａｌｌｂｕｄｄｙにより、アドミニストレータがアルゴリズムを構成して、Ｍｂｕｄｄｙによって実施された重みの割当てを計算するようにすることが可能となる。たとえば、アドミニストレータは現行メトリックの各々に関連づけられた重みを動的に変更しようとすることができる。アドミニストレータは、たとえば、（１）ホスト・メトリックの重みを高くし、（２）サービス・メトリックの重みを低くし、（３）入力メトリックに合わせてエグゼクタ３４０の重みをポーリングする周波数を高くすることを選択できる。Ｃａｌｌｂｕｄｄｙ６２０の構成要素はインタフェース６２２を介してアドミニストレータの要求を受け取り、インタフェース６２４を介してＭｂｕｄｄｙ６１０に通知する。

【００３６】Ｍｂｕｄｄｙ６１０の構成要素は負荷マネージャであり、サーバと接続ルータ・ゲートウェイ・ノ

ードとの間の動的フィードバック制御ループを確立する。Mbuddyはエグゼクタ610の経路指定アルゴリズムの重みを調節するので、負荷メトリックにしたがって軽く負荷がかけられているサーバは、そのタイプの着信TCP接続の大きい部分を受け取る。上記で定義されたような任意のセットの負荷とポリシー・メトリックが与えられると、Mbuddyはその現行のメトリック及びその現行の重みに基づいて、各VEC内の各ポートの各サーバに対する新しい相対重みを計算する。

【0037】あらゆるVECにおける各ポートに対する重みの割当ては次のようにして計算される。（1）すべての実行サーバに対するすべての総合メトリック（AM）を計算する。（2）各実行サーバ（CWP）に対するすべての現行の重みの比率を計算する。（3）各メトリックMに関し、各サーバSについてその値のメトリック比率（MP）を計算する（総合AMに関して）。（4）各サーバに関し、新しい重みNWを計算する。（4a）サーバが静止している場合には、NWを0にセットする。（4b）サーバがスティッキィ重みWを有している場合には、Wの値をNWとして使用する。（4c）次の式によって、ベクトルNWVを計算する（ただし、各項目NWV［i］は単一のメトリックM［i］に基づく）。

NWV［i］=AW+［（CWP－MP）／SF］

ただし、AWは重みの現行範囲内での平均重みであり、SFは平滑か計数のパラメータである。（4d）各サーバの新しい重みNWを次のものとして計算する。

NW=NWV［1］＊W［1］+NWV［2］＊W［2］+．．．+NWV［i］＊W［i］

【0038】図11は本発明により、メトリックがどのようにメトリックによって受け取られるのか、また重みの割当てがどのように計算されるのかについての流れ図の説明である。最上部の枠910はマネージャMbuddyがメッセージまたはタイムアウトのいずれかであるイベントを待機しているところを示す。判断ブロック920はこれらのイベントのタイプを決定する。値のリフレッシュを必要とするタイムアウトがあった場合、ブロック930において、エグゼクタに照会を行って、入力メトリックを与える一連のカウンタ値を取得する（935）。ブロック920で、イベントがパラメータの更新についての要求であると判断された場合、対応するパラメータが更新される（928）。たとえば、アドミニストレータはメトリックに関連づけられた重み、またはポーリング期間を更新することができる。ブロック920で、イベントがメトリック更新の受取りであったと判断された場合には、ブロック925において、メトリックが検索され、これにしたがって、内部変数がセットされる。新しいメトリックが到着した場合には、ブロック940において、アルゴリズムがすべてのメトリックの現行の比率と、現行の重みを計算する。次いで、ブロック950において、サーバ・ノードの各々に対する新しい重みNWが、上述の式を使用して計算される。このブロックは各サーバiが重み項目を有している重みNW［i］の新しいベクトルを作成する。判断ブロック960は任意の閾値機能によって、重みNW［i］の計算された新しいベクトルが、現行の重みベクトルと異なっているかどうかを判断する。新しいベクトルが異なっている場合には、ブロック970において、エグゼクタに新しい重みが通知され、それ以外であれば、アルゴリズムはTOP状態に戻り、新しいイベントを待機する。

【0039】4．回復マネージャ

機能しているゲートウェイの障害を検出すると、指定されたバックアップ・ゲートウェイにおける回復マネージャが活動状態となる。障害検出は従来、A. Bhide他、「A highly Available Network File Server」、USENIX Conference、1991年冬期、テキサス州ダラス、第199ページ、またはF. Jahanian他、「Processor Group Membership Protocols: Specification, Design and Implementation」、Proceedings of the 12th Symposium of Reliable Distributed Systems、第2－11ページ、Princeton、ニュージャージー州、1993年10月、IEEE Computer Societyに記載されているようにして行うことができる。

【0040】回復マネージャはまずHA／NFSで教示されているようにして、障害を起こしたゲートウェイからネットワーク接続を除去し、次いで、すべての活動サーバ・ノードに問い合わせを行って、それぞれのシャドウ接続テーブルから状態情報を取得し、この情報からゲートウェイのメッセージ・スイッチ内に接続テーブルを構成する。引継プロセスはTCP／IPのタイムアウト期間内に完了していなければならず、その結果、既存の接続が失われることはない。これを達成するために、内部ノードは新規なハイブリッド・アルゴリズム（後述）を実行して、接続が非活動状態になった時点を感知し、これらをそれぞれのシャドウ接続テーブルから除去し、その結果、活動接続が引き継ぎゲートウェイに対して記述される。すべての機能中のクラスタ・ノードが応答すると（指定の期間内に応答しないノードは機能していないものと見なされる）、バックアップ・ゲートウェイで実行されている回復マネージャはそれ自体のネットワーク・インタフェースを使用可能とし、クラスタのipアドレス宛のパケットを受け取る。この最後のステップはバックアップ・ゲートウェイが作動するのを可能とするのに必要な作業を完了する。マネージャの構成要素によって使用される相対的に静的な構成データは、主ゲートウェイトバックアップ・ゲートウェイトの間で共用されているファイルに維持され、引継中にバックアップによって読み取られる。

【0041】自明ではあるが、受け入れられない解決策は接続情報をバックアップ・ゲートウェイに重複して維

持することである。これは各確立された接続及び各終了した接続における主ゲートウェイトバックアップ・ゲートウェイの間の「2段階」プロトコルを必要とし、パフォーマンスのコストが厳しいものであるため、拒絶された。

【0042】図12は高可用性のゲートウェイを備えたカプセル化クラスタの構成を示す。主ゲートウェイ1050は外部ネットワーク120に活動状態で接続されている。指定されたバックアップ・ゲートウェイ1030はネットワーク120に対する物理的ではあるが、非活動状態の接続を含んでいる。正規のカプセル化クラスタ・ゲートウェイの構成要素、マネージャ320及びエグゼクタ340に加えて、各ゲートウェイは回復マネージャ1020を含んでいる。（主ゲートウェイは障害及び回復後にバックアップとなることができる。）各サーバ・ノード107はシャドウ接続テーブル1010を含んでおり、このテーブルには、外部ネットワーク120に対するその活動接続に関する情報が維持されている。

【0043】メッセージ（ipパケット）がクラスタ・ゲートウェイに到着し、特定のTCPまたはUDPプロトコル・ポートへ送られる。ゲートウェイ内のメッセージ・スイッチにより、メッセージ経路指定機能をプロトコル・ポートに合わせてインストールすることが可能となる。経路指定機能が関連づけられたポートに到着する各メッセージに対して呼び出され、内部ノード及びメッセージが送られるポートの選択を担当する。確立済みの接続を指定する情報及び接続を保持するクラスタが、ゲートウェイ内のテーブルに記録される。このテーブルをメッセージ・スイッチが使用して、確立済みの接続上の着信パケットを適正なクラスタ・ノードへ送る。

【0044】どのサーバ・ポートがインストールされたメッセージ・スイッチ機能を持っているかなどの相対的に静的な情報が維持され、他のマネージャ構成情報が、主ゲートウェイ及びバックアップ・ゲートウェイ両方にアクセス可能な共用ファイルに保管される。現行の接続情報はきわめて迅速に変化し、本明細書に記載する技法にしたがって管理される。

【0045】各内部ノードF107はそれ自体の接続のために（他のノードに対する接続に対してではなく）ゲートウェイ経路指定テーブルのシャドウ1010を維持している。このシャドウ・テーブルをノードが使用して、引継中にバックアップ・ゲートウェイ1030における回復マネージャ1020からの、引継ゲートウェイの要求に応答する。このテーブルは内部ノードが引継ゲートウェイに応答するのに必要とする時間を大幅に少なくし、これは確立済みの接続を活動させておくためには、接続ベースのプロトコルが通信を正常に完了できるようにする「タイムアウト」期間内に引継ゲートウェイが作動していなければならないため、きわめて重要である。

【0046】接続テーブルの項目に対するスペースを再利用するため、ゲートウェイ内、及び内部ノードに維持されているシャドウ内の両方で、次のように進める。接続は2つの状態、すなわちACTIVEまたはFINのうちいずれかになっている。接続テーブル項目には参照ごとにタイム・スタンプがつけられる。FIN_TIME_OUTと呼ばれるユーザ構成可能なタイマが維持されている。このタイマは閉鎖されると考えられるFIN状態における会話に対する最後の参照後の時点を表している。タイマはサービス・アドレスごと、あるいはポートごとのグローバルなものでよい。活動クローズ（接続の一方の側がFINを送ったが、他方が接続上で送信を継続している）の意図は、サーバがクライアントへのデータの送信を継続することができ、データ電送の終了時に、会話が閉鎖されることである。クライアントはサーバにクライアントからこれ以上要求が送られないということをサーバに伝える手段として会話を能動的に閉鎖する。ここでの検討のために、クライアントの要求がルータを経由して送られているものと想定する。このプロトコルが機能するのは、サーバが肯定応答を受けるデータを送り続けているからである。ルータは、したがってサーバは肯定応答を見て、接続テーブル項目に連続的にタイム・スタンプをつける。サーバがクライアントに対するデータの送信を完了し、その会話の「半分」を閉鎖すると、最後の肯定応答がクライアントからサーバへ向かって流れる。FIN_TIME_OUTが経過した後、サーバは接続項目をパージすることができる。第2のタイマSTALE_TIME_OUTがゲートウェイによって維持されている。STALE_TIME_OUT以外にもはや何の活動もしていない活動状態の何らかの接続はパージできる。

【0047】このアルゴリズム（接続再構成アルゴリズム）は、バックアップゲートウェイによる引継プロセスのサポートを内部ノードが維持している接続テーブルのシャドウ内の項目に対するスペースを再利用するために、内部ノードにおいても実行される。このようにして、シャドウ・テーブル内の項目の数をできるだけ少ない数に維持し、これにより引き継ぎプロセスをできるだけ迅速に進めることができる。

【0048】デフォルトとして、FIN_TIME_OUTをTCPの最小セグメント超（MSL）の3倍の値にセットする必要がある。デフォルトのSTALE_TIME_OUTはTCPの失効タイムアウトよりも長くなければならない。タイマが関連づけられているプロトコルを考慮することにより、FIN_TIME_OUTに対するより妥当な値を見つけだすことができる。

【0049】主ゲートウェイが障害を起こしたと判断されたか、あるいは明示のアドミニストレーション上の判断によって判断されたかにかかわらず、バックアップ・ゲートウェイが活動状態とならなければならないと判断された場合、以下のステップがバックアップ・ゲートウェ

イ１０３０内の回復マネージャ１０２０によって取られる。

【００５０】（１）ｉｐアドレス引継を使用して、バックアップ・ゲートウェイは主ゲートウェイのネットワーク接続を除去する。ダウンしたと想定されるゲートウェイが実際に、ネットワークからのメッセージを受け入れないようにするため、このステップが必要である。このステップを行わないと、ある種の障害、すなわち「部分的に障害を起こしたゲートウェイ」がメッセージの受取りを継続し、処理を行うという障害が生じる可能性があり、これはシステムの保全性を危険にさらすものである。

【００５１】（２）バックアップ・ゲートウェイがクラスタの機能している各ノードに問い合わせを行い、それぞれのノードに割り振られているすべてのＵＤＰポート、主ゲートウェイクラスタ外のホストとの間の主ゲートウェイを介して確立されたＴＣＰ接続の記述を要求する。バックアップ・ゲートウェイは私用ｉｐベース・プロトコルを使用してこれを行う。各ノードに維持されているシャドウ接続テーブルはノードからの即時応答を可能とし、確立済み接続がゲートウェイの引継時にタイムアウトとならない確率を高くする。閉鎖した接続を認識し、これらをサポートするために使用されるスペースを再利用するための上述したアルゴリズムは、シャドウ接続テーブルのサイズを最小限度のものとし、ゲートウェイの引継を達成するのに必要な時間を短縮するのに寄与する。

【００５２】（３）バックアップ・ゲートウェイが機能している各ノードからの応答を記録し、ノードのＵＤＰポート及びＴＣＰ接続を、図１２のバックアップ・ゲートウェイのエグゼクタ３４０内の図５の接続テーブル５１０に記録する。

【００５３】（４）すべての機能しているクラスタ・ノードが応答した場合（指定された期間内に応答しないノードは機能していないものと見なされる）、バックアップ・ゲートウェイはそれ自体のネットワーク・インタフェースを使用可能とし、クラスタのｉｐアドレス宛のパケットを受け取るようにする。この最後のステップはバックアップ・ゲートウェイが作動状態となることを可能とするのに必要な作業を完了する。

【００５４】本発明を好ましい実施の形態により説明してきたが、各種の改変形及び改善が当分野の技術者には思い浮かぶであろう。それ故、好ましい実施の形態が例として挙げられたものであって、限定事項として挙げられたものではないことを理解すべきである。本発明の範囲は首記の特許請求の範囲によって画定されるものである。

【００５５】まとめとして、本発明の構成に関して以下の事項を開示する。

【００５６】（１）コンピュータ・ノードのクラスタの境界を越えて着信メッセージを経路指定する方法であって、前記クラスタが１つまたは複数のネットワークに結合されており、前記方法がポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号及び目標アドレスを見つけだし、読み取るステップと、前記目標アドレスに基づいて、前記コンピュータ・ノードのサブセットを選択するステップと、前記ポート番号に基づいて、前記サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定する機能を選択するステップと、前記メッセージを前記経路の宛先に送るステップと、前記メッセージが境界を越えて送られている間に、前記サブセット内のメンバシップの少なくとも１つと前記サブセットの数を動的に変更するステップとを備えている方法。

（２）選択が前記ポート番号と前記メッセージ・ヘッダ内のプロトコル識別子に基づいていることを備えている、上記（１）に記載の方法。

（３）前記サブセットの数が監視機能によって動的に変更される、上記（１）に記載の方法。

（４）前記サブセットのメンバシップが監視機能によって動的に変更される、上記（１）に記載の方法。

（５）変更が前記サブセットのメンバの置換、及び前記サブセットへのメンバの追加の少なくとも１つを含む、上記（３）に記載の方法。

（６）コンピュータ・ノードのクラスタを越えて着信メッセージを透過的に経路指定するシステムであって、前記クラスタが１つまたは複数のネットワークに結合されており、前記システムがポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号及び目標アドレスを見つけだし、読み取って、前記コンピュータ・ノードのサブセットを選択する手段と、前記ポート番号に基づいて、機能を選択し、該機能が前記サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定し、該経路の宛先が前記サブセット内の前記コンピュータ・ノードである手段と、前記サブセット内のメンバシップの少なくとも１つと前記サブセットの数を動的に変更する手段とを備えているシステム。

（７）前記サブセットの数が監視機能によって動的に変更される、上記（６）に記載のシステム。

（８）前記サブセットのメンバシップが監視機能によって動的に変更される。上記（７）に記載のシステム。

（９）変更が前記サブセットのメンバの置換、及び前記サブセットへのメンバの追加の少なくとも１つを含む、上記（７）に記載のシステム。

（１０）コンピュータ・ノードのクラスタの境界を越えて着信メッセージを経路指定する方法であって、前記クラスタが１つまたは複数のネットワークに結合されており、前記方法が境界ノードにおいて、ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号を見つけだし、読み取るステップと、前記ポート番号に基づい

て、サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定し、該経路の宛先が前記クラスタ内のコンピュータ・ノードであるステップとを備えており、前記境界ノードの障害を検出するステップと、障害の検出に応答して、状態情報のサブセットを前記クラスタ内の各ノードから代替境界ノードへ転送するステップと、前記代替境界ノードにおいて、前記クラスタ内のノードから状態情報のサブセットを収集するステップと、前記状態情報を使用して、障害前の前記境界ノードの作動状態を再構成し、メッセージが障害前に前記境界ノードによって達成されていたのと同じ態様で、前記代替境界ノードによって分配されるようにするステップとを備えている方法。

（11）ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号及び目標アドレスを見つけだし、読み取るステップと、前記目標アドレスに基づいて、前記コンピュータ・ノードのサブセットを選択するステップとをさらに備えており、経路の宛先が前記サブセットから選択される、上記（10）に記載の方法。

（12）コンピュータ・ノードのクラスタの境界ノードの障害から回復するシステムであって、前記境界ノードにおいて、ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号を見つけだし、読取り、前記ポート番号に基づいて、サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定する機能を選択し、前記経路の宛先が前記クラスタ内の前記コンピュータ・ノードである手段と、前記境界ノードの障害を検出する手段と、障害の検出に応答して、前記クラスタ内の各ノードから状態情報のサブセットを収集する手段と、前記サブセットから、障害前の前記境界ノードの作動状態を再構成し、メッセージが障害前に前記境界ノードによって達成されていたのと同じ態様で、代替ノードによって分配されるようにする手段とを備えている代替境界ノードとを備えているシステム。

（13）ポート・タイプ・メッセージのメッセージ・ヘッダ内のポート番号を見つけだし、読取り、前記ポート番号に基づいて、サブセット内の複数の考えられる宛先からメッセージに対する経路の宛先を判定する機能を選択し、前記経路の宛先がクラスタ内のコンピュータ・ノードである手段と、障害の検出に応答して、前記クラスタ内の各ノードから状態情報のサブセットを収集する手段と、サブセットから、障害前の境界ノードの作動状態を再構成し、メッセージが障害前に前記境界ノードによって達成されていたのと同じ態様で、代替ノードによって分配されるようにする手段とを備えているコンピュータ・ノードのクラスタで使用される境界ノード。

【図面の簡単な説明】

【図1】従来技術のカプセル化クラスタ・システムを示す図である。

【図2】従来技術のメッセージ・スイッチを示す図である。

【図3】本発明の実施の形態による仮想カプセル化クラスタ・システムを示す図である。

【図4】本発明の他の実施の形態による仮想カプセル化クラスタ・システムを示す図である。

【図5】図3及び図4のエグゼクタの詳細な図である。

【図6】図3及び図4のマネージャの詳細な図である。

【図7】エグゼクタの流れ図である。

【図8】エグゼクタの流れ図である。

【図9】エグゼクタの流れ図である。

【図10】エグゼクタのデータ構造を示す図である。

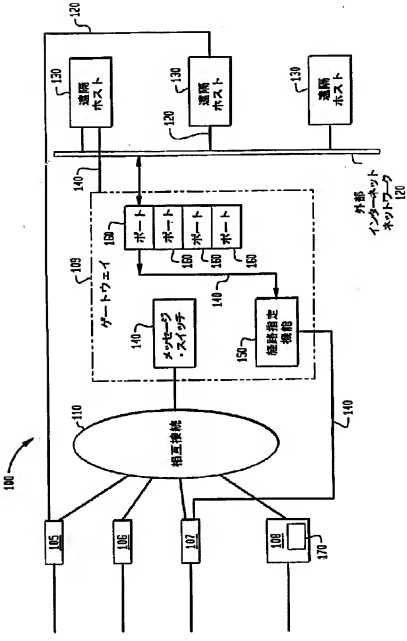
【図11】マネージャの流れ図である。

【図12】本発明の実施の形態による高可用性ゲートウェイを有するクラスタを示す図である。

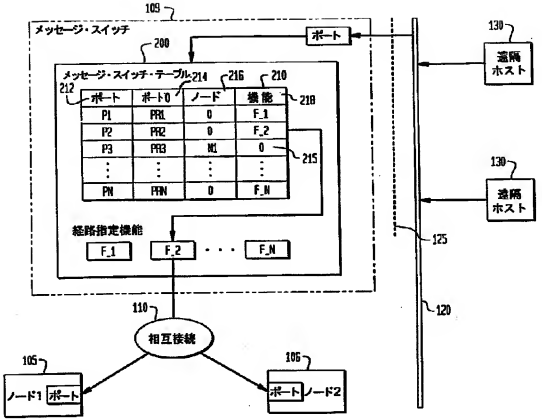
【符号の説明】

105　コンピュータ・ノード
106　コンピュータ・ノード
107　コンピュータ・ノード
108　コンピュータ・ノード
109　ゲートウェイ
110　相互接続
120　ネットワーク
140　メッセージ・スイッチ
300　TCP接続ルータ（TCP－CR）
310　VECルータ
320　マネージャ
340　エグゼクタ

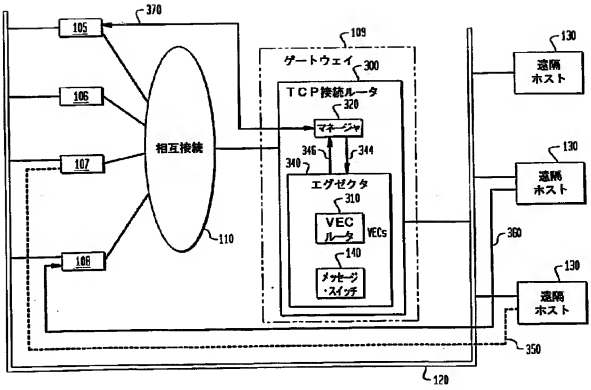
【図1】


120
130
通信ホスト
130
通信ホスト
130
通信ホスト
120
140
外部インターネットネットワーク120
ゲートウェイ
109
150
ポート
ポート
ポート
ポート
160
160
160
160
140
140
メッセージスイッチ
150
経路指定機能
110
相互接続機
140
100
105
106
107
108
170

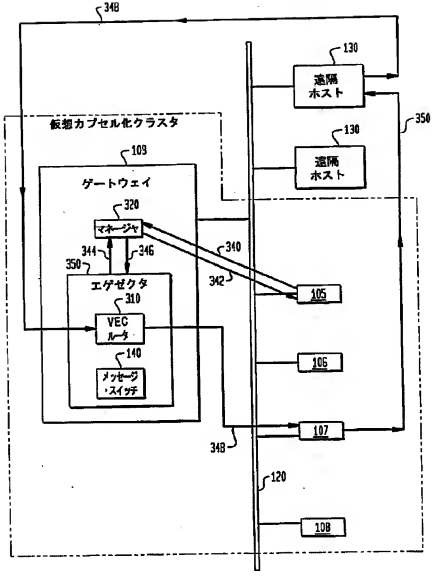
【図２】


109
メッセージ・スイッチ
ポート
200
メッセージ・スイッチ・テーブル
214
216
210
212
ポート
ポートID
ノード
機能
218
P1
PR1
0
F_1
P2
PR2
0
F_2
P3
PR3
NL
0
215
PN
PRN
0
F_N
経路指定機能
F_1
F_2
F_N
110
相互接続
105
ノード1
ポート
105
ポート
ノード2
130
遠隔ホスト
130
遠隔ホスト
125
120

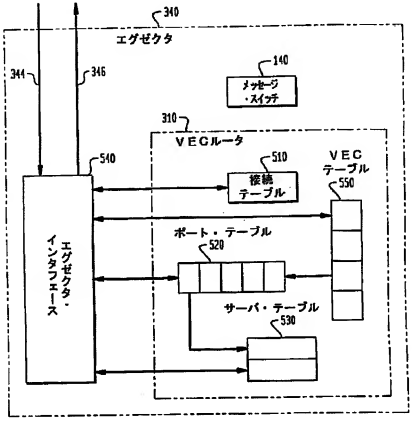
【図3】


105
370
106
107
108
相互接続
110
109
ゲートウェイ
300
TCP接続ルータ
320
マネージャ
346
344
340
エグゼクタ
310
VEC
ルータ
VECs
140
メッセージ
・スイッチ
130
遠隔
ホスト
130
遠隔
ホスト
130
遠隔
ホスト
360
350
120

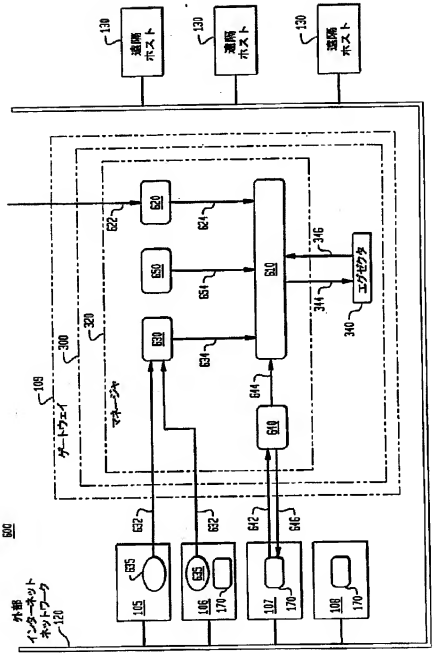
【図4】


348
130
遠隔
ホスト
350
130
遠隔
ホスト
仮想カプセル化クラスタ
109
ゲートウェイ
320
マネージャ
340
344
346
350
342
エゲゼクタ
310
105
VEC
ルータ
106
140
メッセージ
・スイッチ
107
348
120
108

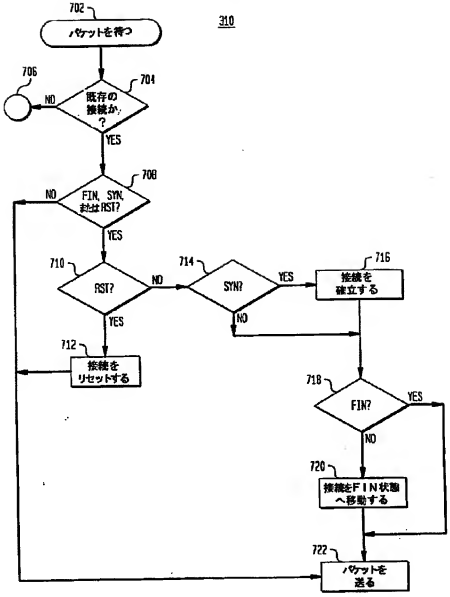
【図5】


340
エグゼクタ
344
346
140
メッセージ・スイッチ
310
VECルータ
540
510
接続テーブル
VEC テーブル
550
ポート・テーブル
520
エグゼクタ・インタフェース
サーバ・テーブル
530

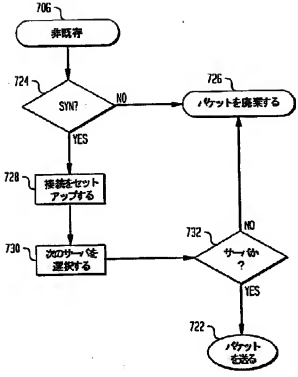
【図6】


遠隔ホスト
130
遠隔ホスト
130
遠隔ホスト
130
ゲートウェイ
100
300
320
マネージャ
620
650
630
610
622
624
651
634
641
346
344
340
コンタクタ
632
642
646
635
645
105
106
107
108
170
外部インターネットネットワーク
120
600

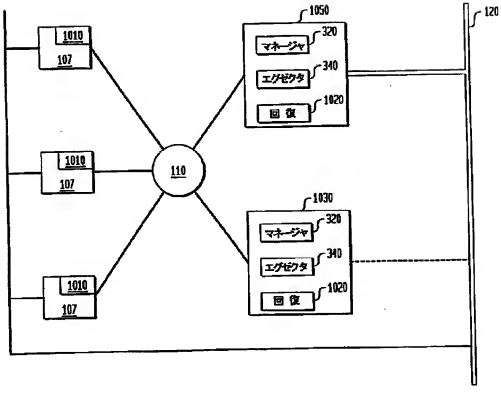
【図7】


310
702
パケットを待つ
704
既存の接続か？
706
NO
YES
708
FIN、SYN、またはRST?
NO
YES
710
RST?
NO
YES
712
接続をリセットする
714
SYN?
YES
NO
716
接続を確立する
718
FIN?
YES
NO
720
接続をFIN状態へ移動する
722
パケットを送る

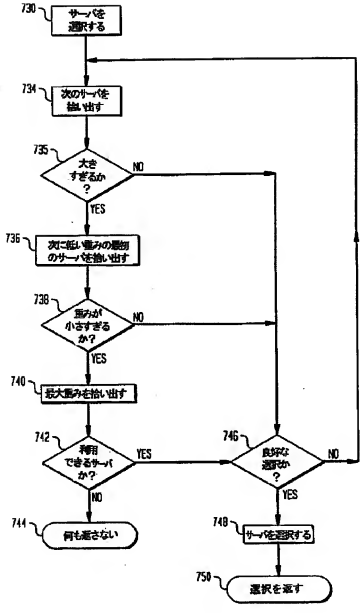
【図８】


706
非既存
724
SYN?
NO
726
パケットを廃棄する
YES
728
接続をセットアップする
730
次のサーバを選択する
732
NO
サーバか？
YES
722
パケットを送る

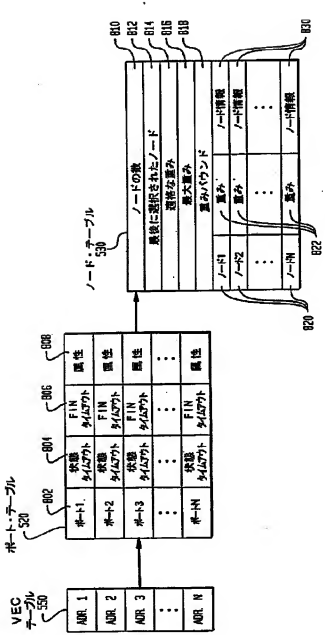
【図１２】


1010
107
1010
107
1010
107
110
1050
マネージャ
320
エグゼクタ
340
回復
1020
1030
マネージャ
320
エグゼクタ
340
回復
1020
120

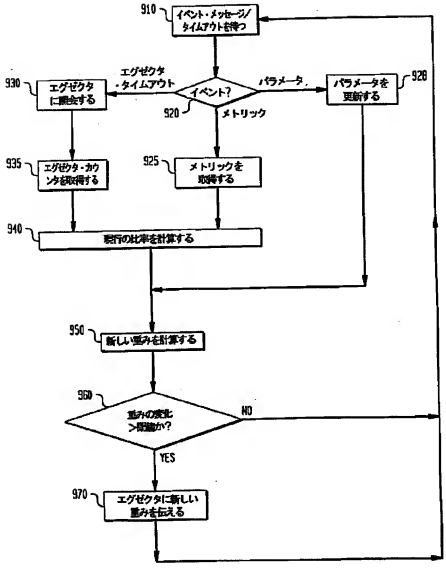
【図９】


730
サーバを選択する
734
次のサーバを拾い出す
735
大きすぎるか？
NO
YES
736
次に近い重みの最初のサーバを拾い出す
738
重みが小さすぎるか？
NO
YES
740
最大重みを拾い出す
742
利用できるサーバか？
YES
NO
744
何も返さない
746
良好な選択か？
NO
YES
748
サーバを選択する
750
選択を返す

【図10】


VEC
テーブル
550
ADR. 1
ADR. 2
ADR. 3
ADR. N
ポート・テーブル
520
802
ポート1
ポート2
ポート3
ポートN
状態
タイムアウト
804
FIN
タイムアウト
806
属性
808
ノード・テーブル
530
ノードの数
最後に選択されたノード
選択な重み
最大重み
重みバウンド
ノード1
ノード2
ノードN
重み
ノード情報
810
812
814
816
818
820
822
830

【図１１】



フロントページの続き

(72)発明者　ジャーマン・セルジオ・ゴールドシュミット
アメリカ合衆国10522　ニューヨーク州ドッブス・フェリー　チェストナット・リッジウェイ　21

(72)発明者　ガーニイ・ダグラス・ホロウェイーハント
アメリカ合衆国12590　ニューヨーク州ワッピンジャー・フォールス　エッジヒル・ドライブ　127

(72)発明者　ステファン・エドウィン・スミス
アメリカ合衆国10541　ニューヨーク州マホパックハットフィールド・ロード　19